ZMAC: Specification, Security Proof, and Instantiation Updates*

Tetsu Iwata†

Nagoya University, Japan

Joint work with Kazuhiko Minematsu, Thomas Peyrin, and Yannick Seurin

ASK 2017
Fenglin Hotel, Changsha, China
December 10, 2017

* Based on: Iwata, Minematsu, Peyrin, and Seurin. ZMAC: A Fast Tweakable Block Cipher Mode for Highly Secure Message Authentication. CRYPTO 2017

† Supported by JSPS KAKENHI, Grant-in-Aid for Scientific Research (B), Grant Number 26280045
Introduction: Message Authentication Code (MAC)

- Symmetric-key Crypto for tampering detection
- $\text{MAC} : \mathcal{K} \times \{0, 1\}^* \rightarrow \mathcal{T}$
- Alice computes $\text{Tag} = \text{MAC}(K, M) = \text{MAC}_K(M)$ and sends $(M, \text{Tag})$ to Bob
- Bob checks if $(M, \text{Tag})$ is authentic by computing tag locally
- If $\text{MAC}_K(*)$ is a variable-input-length PRF, it is secure
Tweakable Block Cipher (TBC)

Extension of ordinal Block Cipher (BC), formalized by Liskov et al. [LRW02]

- \( \tilde{E} : K \times T \times M \rightarrow M \), tweak \( T \in T \) is a public input
- \( (K, T) \in K \times T \) specifies a permutation over \( M \)
- Let \( M = \{0, 1\}^n \) and \( T = \{0, 1\}^t \)

We implicitly assume additional small tweak \( i = 1, 2, \ldots \), used for domain separation, and write as \( \tilde{E}_K^i(T, X) \) when necessary
Building TBC

Block cipher modes for TBC: LRW [LRW02] and XEX [Rog04]

- Efficient but security is up to the birthday bound ($O(2^{64})$ attack when AES is used)
- Beyond-the-birthday-bound (BBB) security is possible (e.g. [Min09][LST12][LS15]) but not really efficient

Dedicated designs:

- HPC [Sch98]
- Threefish in Skein hash function [FLS+10]
- Deoxys-BC, Joltik-BC, KIASU-BC [JNP14a], SCREAM [GLS+14],
  - in the CAESAR submissions
- SKINNY [BJK+16], QARMA [Ava17], ...
Security notions of TBC [LRW02]

- Indistinguishable from the set of independent uniform random permutations indexed by tweak
  - Tweakable uniform random permutation (TURP) denoted by $\tilde{P}$
  - Tweak is chosen by the adversary
- CCA-secure TBC = TSPRP
Security notions of TBC [LRW02]

- Indistinguishable from the set of independent uniform random permutations indexed by tweak
  - Tweakable uniform random permutation (TURP) denoted by \( \tilde{P} \)
  - Tweak is chosen by the adversary

- CCA-secure TBC = TSPRP
- CPA-secure TBC = TPRP
Building MAC with TBC : PMAC1

PMAC1 by Rogaway [Rog04], introduced in the proof of PMAC

- Parallel
- Security is up to the birthday bound wrt the block size \(n\)
  - \(\text{Adv}_{\text{PMAC1}}^{\text{tprp}}(\sigma) = O(\sigma^2/2^n)\) for \(\sigma\) queried blocks
  - Thus \(n/2\)-bit security
Building MAC with TBC: PMAC_TBC1k

PMAC_TBC1k by Naito [Nai15]

- $2n$-bit chaining similar to PMAC_Plus [Yas11]
  - Finalization by $2n$-bit PRF built from TBC
- BBB-secure: improve security of PMAC1 to $n$ bits
- Same computation cost as PMAC1 (except for the finalization)

PMAC_TBC1k (message hashing part)
Efficiency of MAC

These TBC-based MACs are **not** optimally efficient

- They process $n$-bit input per 1 TBC call
- $t$-bit tweak does not process message – reserved for block index
Efficiency of MAC

These TBC-based MACs are **not** optimally efficient

- They process *n*-bit input per 1 TBC call
- *t*-bit tweak does not process message – reserved for block index

Optimally-efficient TBC-based MAC?
Our proposal: ZMAC (“The MAC”) [IMPS17]

ZMAC is

- The first **optimally efficient** TBC-based MAC
  - \((n + t)\)-bit input per 1 TBC call
- Parallel, and **BBB-secure**
  - \(\min\{n, (n + t)/2\}\)-bit security, e.g. \(n\)-bit-secure when \(t \geq n\)

It uses TBC as a sole primitive, and secure if TBC is a TPRP
Structure of ZMAC

A simple composition of message hashing and finalization (Carter-Wegman MAC):

- $\text{ZMAC} = \text{ZFIN} \circ \text{ZHASH}$
- $\text{ZHASH} : M \rightarrow \{0, 1\}^{n+t}$ is a computational universal hash function
- $\text{ZFIN} : \{0, 1\}^{n+t} \rightarrow \{0, 1\}^{2n}$ is a PRF
  - Output truncation if needed

Unified specs for any $t$ ($t = n$ or $t < n$ or $t > n$)
Structure of ZMAC

A simple composition of message hashing and finalization (Carter-Wegman MAC):

- \( ZMAC = ZFIN \circ ZHASH \)
- \( ZHASH : \mathcal{M} \rightarrow \{0, 1\}^{n+t} \) is a computational universal hash function
- \( ZFIN : \{0, 1\}^{n+t} \rightarrow \{0, 1\}^{2n} \) is a PRF
  - Output truncation if needed

Unified specs for any \( t \) (\( t = n \) or \( t < n \) or \( t > n \))

We focus on \( ZHASH \)
How ZHASH works: tweak extension

Optimal efficiency implies $t$-bit tweak of $\tilde{E}$ must be extended to incorporate block index
This can be done by XTX [MI15], an extension of LRW and XEX:

- Global tweak $G \in \mathcal{G}$, $|\mathcal{G}| > 2^t$
- Keyed function $H : \mathcal{L} \times \mathcal{G} \rightarrow (\{0, 1\}^n \times \{0, 1\}^t)$
- $\text{XTX}[\tilde{E}, H]_{K,L}(G, X) = \tilde{E}_K(W_t, W_n \oplus X) \oplus W_n$ with $(W_n, W_t) = H_L(G)$
How ZHASH works: security of XTX/XT

XTX is secure if $H$ is $\epsilon$-partial AXU (pAXU) [MI15]:

$$\max_{G \neq G', \delta \in \{0,1\}^n} \Pr[L \leftarrow \mathcal{L} : H_L(G) \oplus H_L(G') = (\delta, 0^t)] \leq \epsilon$$

that is, $n$-bit part is close to differentially uniform and $t$-bit part has a small collision probability
How ZHASH works: security of XTX/XT

In our case, \( G \in \{0, 1\}^t \times \mathbb{N} \), and block index is a counter

Then XTX can be instantiated and optimized by

- Using the “doubling” trick as XEX
- Omitting the outer mask to \( Y \) (as decryption is not needed)

\( \dagger \) Omitting domain separation variable
How ZHASH works: security of XTX/XT

The resulting scheme is $\text{XT}$, using $H_L(G)$ defined as

$$H_{(L_\ell,L_r)}(T,i) = \left(2^{i-1}L_\ell, 2^{i-1}L_r \oplus_t T\right),$$

using two $n$-bit keys $(L_\ell, L_r)$

Details:

- $2^i X$ is $X$ multiplied by 2 over $\text{GF}(2^n)$ for $i$ times
  - Computation is easy by caching $2^{i-1}X$ as done in XEX
- $X \oplus_t Y = \text{msb}_t(X) \oplus Y$ if $t \leq n$, $(X \parallel 0^{t-n}) \oplus Y$ if $t > n$
  - Chop-or-pad before sum
How ZHASH works: security of XTX/XT

Lemma

Let $\tilde{P} : \mathcal{T} \times \{0, 1\}^n \rightarrow \{0, 1\}^n$ be a TURP and $H$ is $\epsilon$-pAXU. Then,

$$\text{Adv}^{\text{tprp}}_{\text{XT}[\tilde{P}, H]}(q) \leq \frac{q^2 \epsilon}{2}.$$

and our $H$ is $1/2^{n+\min\{n,t\}}$-pAXU. Thus,

$$\text{Adv}^{\text{tprp}}_{\text{XT}[\tilde{P}, H]}(q) \leq \frac{q^2}{2^{n+\min\{n,t\}+1}}.$$

Therefore, XT has $\min\{n, (n + t)/2\}$-bit, BBB-security.
How ZHASH works: chaining scheme

Given XT, it’s easy to apply it in the PMAC-like single-chaining hashing scheme

- Message is divided into \((n + t)\)-bit blocks, \((X_\ell[i], X_r[i])\) for \(i = 1, 2, \ldots\)
- This is optimally efficient, but security is up to the birthday bound
How ZHASH works: chaining scheme

Given XT, it’s easy to apply it in the PMAC-like single-chaining hashing scheme

- Message is divided into \((n + t)\)-bit blocks, \((X_\ell[i], X_r[i])\) for \(i = 1, 2, \ldots\)
- This is optimally efficient, but security is up to the birthday bound
- Need a larger chaining value
How ZHASH works: chaining scheme

- Naive use of $2^n$-bit chaining scheme [Nai15][Yas11] doesn’t work
  - XT output collision still breaks the scheme
How ZHASH works: chaining scheme

- Key observation: to avoid these collision attacks, the process of \((X_\ell, X_r)\) (the dotted box) must be a permutation.
- A Feistel-like 1-round permutation works (ZHASH).

![Diagram of ZHASH](attachment:diagram.png)
How ZHASH works: chaining scheme

- Key observation: to avoid these collision attacks, the process of \((X_\ell, X_r)\) (the dotted box) must be a permutation
- A Feistel-like 1-round permutation works (ZHASH)

**Lemma**

ZHASH (w/ XT using TURP) is \(\epsilon\)-almost universal for \(\epsilon = 4/2^n + \min\{n,t\}\)
Full ZHASH

Input: \( X = (X[1], \ldots, X[m]) \), \(|X[i]| = n + t\)

Output \((U, V)\), \(|U| = n\), \(|V| = t\)

Details:

- \( X \oplus_t Y = \text{msb}_t(X) \oplus Y \) if \( t \leq n \), \((X \parallel 0^{t-n}) \oplus Y \) if \( t > n \)
- \( 2 \cdot X \): multiplication by 2
- \( L_\ell \) and \( L_r \): two \( n\)-bit masks from \( \tilde{E}_K \) w/ domain separation
ZFIN

ZFIN simply encrypts $U$ with tweak $V$ twice (for each $n$-bit output) and takes a sum (with domain separation)

\[
\tilde{E}_i K U V \oplus \tilde{E}_{i+1} K U V \oplus \tilde{E}_{i+2} K U V \oplus \tilde{E}_{i+3} K U V
\]

PRF security of ZFIN

- ZFIN is essentially “Sum of Permutations” [Luc00, BI99, Pat08a, Pat13, CLP14, MN17]
- From a recent result by Dai et al. [DHT17], ZFIN is \textit{n-bit secure}

\[
\text{Adv}_{ZFIN[P]}^{\text{prf}}(q) \leq 2 \left( \frac{q}{2^n} \right)^{3/2}
\]
Security of ZMAC

Combining all lemmas,

**Theorem**

For $q \leq 2^{n-4}$ queries of total $\sigma (n + t)$-bit blocks,

$$\text{Adv}^{\text{prf}}_{\text{ZMAC}[\tilde{P}]}(q, \sigma) \leq \frac{2.5\sigma^2}{2^n + \min\{n,t\}} + 4 \left(\frac{q}{2^n}\right)^{3/2}.$$ 

Thus ZMAC is $\min\{n, (n + t)/2\}$-bit secure
• **ZHASH** is $\epsilon$-almost universal for $\epsilon = 4/2^{n+\min\{n,t\}}$

• $\max_{X \in \{0,1\}^{n+t}} \Pr_{XT} [\text{ZHASH}_{XT}(X) = \text{ZHASH}_{XT}(X')] \leq \epsilon$

\[\text{ZHASH}\]
A Feistel-like Network Is a Permutation

- red lines are $t$ bits
- $X \oplus_t Y = \text{msb}_t(X) \oplus Y$ if $t \leq n$, $(X \parallel 0^{t-n}) \oplus Y$ if $t > n$
Breaking into Cases

- **ZHASH** is $\epsilon$-almost universal for $\epsilon = \frac{4}{2^n + \min\{n, t\}}$
- For any distinct $X \in (\{0, 1\}^{n+t})^m$ and $X' \in (\{0, 1\}^{n+1})^{m'}$,
  \[
  \Pr_{X,T}[\text{ZHASH}_{XT}(X) = \text{ZHASH}_{XT}(X')] \leq \epsilon
  \]

Cases:

1. $m = m'$, $\exists h, X[h] \neq X'[h]$, and $\forall i \neq h, X[i] = X'[i]$
   (same number of blocks, difference in exactly one block)

2. $m = m'$, $\exists h, s, X[h] \neq X'[h]$ and $X[s] \neq X'[s]$
   (same number of blocks, difference in two (or more) blocks)

3. $m' = m + 1$

4. $m' \geq m + 2$

- focus on the case $t \leq n$
Case 1

- \( m = m', \exists h, X[h] \neq X'[h], \text{ and } \forall i \neq h, X[i] = X'[i] \)
- same number of blocks, difference in exactly one block

\[
\begin{align*}
\Delta X_\ell[h] & \quad \Delta X_r[h] \\
\Delta C_\ell[h] & \quad \Delta C_r[h] \\
& \quad \Delta V \\
& \quad \Delta U
\end{align*}
\]

- \((\Delta C_\ell[h], \Delta C_r[h]) \neq (0^n, 0^t), \text{ so } (\Delta U, \Delta V) \neq (0^n, 0^t)\)
- \(\text{Pr}_{\text{XT}}[\text{ZHASH}_{\text{XT}}(X) = \text{ZHASH}_{\text{XT}}(X')] = 0\)
Case 2

- $m = m', \exists h, s, X[h] \neq X'[h]$ and $X[s] \neq X'[s]$
- same number of blocks, difference in two (or more) blocks

\[\Delta X_{\ell}[h] \quad \Delta X_r[h] \quad \Delta X_{\ell}[s] \quad \Delta X_r[s]\]

\[\Delta C_{\ell}[h] \quad \Delta C_r[h] \quad \Delta C_{\ell}[s] \quad \Delta C_r[s]\]

\[\Delta V \quad \Delta U\]

- $(\Delta C_{\ell}[h], \Delta C_r[h]) \neq (0^n, 0^t)$ and $(\Delta C_{\ell}[s], \Delta C_r[s]) \neq (0^n, 0^t)$
- approach: use $\Delta C_{\ell}[h]$ and $\Delta C_{\ell}[s]$ as randomness
Case 2

\[ \Delta X_\ell[h] \quad \Delta X_r[h] \]

\[ \Delta X_\ell[s] \quad \Delta X_r[s] \]

\[ \Delta C_\ell[h] \quad \Delta C_r[h] \]

\[ \Delta C_\ell[s] \quad \Delta C_r[s] \]

\[ \Delta V \]

\[ \Delta U \]

- \( \Delta U = 0^t \iff 2^{m-h-1} \Delta C_\ell[h] \oplus 2^{m-s-1} \Delta C_\ell[s] = \Delta_1 \)
- \( \Delta V = 0^n \iff \Delta C_r[h] \oplus \Delta C_r[s] = \Delta_2 \)
  \[ \iff \text{msb}_t(\Delta C_\ell[h] \oplus \Delta C_\ell[s]) = \Delta'_2 \]
  \[ \iff \Delta C_\ell[h] \oplus \Delta C_\ell[s] = \Delta'_2 \parallel * \]
Case 2

\[
\begin{align*}
\Delta U = 0^t & \quad \iff \quad 2^{m-h-1} \Delta C_\ell[h] \oplus 2^{m-s-1} \Delta C_\ell[s] = \Delta_1 \\
\Delta V = 0^n & \quad \iff \quad \Delta C_\ell[h] \oplus \Delta C_\ell[s] = \Delta'_2 \parallel * 
\end{align*}
\]

- For each \((\Delta_2, \Delta'_2 \parallel *)\), one possibility for \((\Delta C_r[h], \Delta C_r[s])\)
  - at most \(2^{n-t}\) possible values of \((\Delta C_r[h], \Delta C_r[s])\)
    - s.t. \((\Delta U, \Delta V) = (0^n, 0^t)\)

- at least \((2^n - 1)^2\) possible choices for \((\Delta C_r[h], \Delta C_r[s])\)

- \(\Pr[(\Delta U, \Delta V) = (0^n, 0^t)] \leq \frac{2^{n-t}}{(2^n - 1)^2} \leq \frac{4}{2^{n+t}}\)
Case 3

- \( m' = m + 1 \)

- \( \Delta U = 2(C_{\ell}[m] \oplus 2C'_{\ell}[m] \oplus C'_{\ell}[m + 1] \oplus \Delta_1) \)

- \( \Delta V = \text{msb}_t(C_{\ell}[m] \oplus C'_{\ell}[m] \oplus C'_{\ell}[m + 1]) \oplus \Delta_2 \)

- Use \( C_{\ell}[m], C'_{\ell}[m], C'_{\ell}[m + 1] \) as randomness
Case 3

- $\Delta U = 2(C_\ell[m] \oplus 2C'_\ell[m] \oplus C'_\ell[m + 1] \oplus \Delta_1)$
- $\Delta V = \text{msbt}(C_\ell[m] \oplus C'_\ell[m] \oplus C'_\ell[m + 1]) \oplus \Delta_2$
- $\begin{cases} \Delta U = 0^t \\ \Delta V = 0^n \end{cases} \iff \begin{cases} C_\ell[m] \oplus 2C'_\ell[m] \oplus C'_\ell[m + 1] = \Delta'_1 \\ C_\ell[m] \oplus C'_\ell[m] \oplus C'_\ell[m + 1] = \Delta_2 \parallel * \end{cases}$
- Letting $Y = C_\ell[m] \oplus C'_\ell[m + 1]$ and $Z = C'_\ell[m]$ yields
  \[
  \begin{cases} Y \oplus 2Z = \Delta'_1 \\ Y \oplus Z = \Delta_2 \parallel * \end{cases}
  \]
  which has a unique solution
- they are uniform over $\{0, 1\}^n$
- $\Pr[(\Delta U, \Delta V) = (0^n, 0^t)] \leq \frac{2^{n-t}}{2^{2n}} \leq \frac{1}{2^{n+t}}$
Case 4

- $m' \geq m + 2$

- use $C'_l[m' - 1]$ and $C'_l[m']$ as randomness
- $\Delta U = 2(2C'_l[m' - 1] \oplus C'_l[m'] \oplus \Delta_1)$
- $\Delta V = \text{msb}_t(C'_l[m' - 1] \oplus C'_l[m']) \oplus \Delta_2$
- the same analysis as Case 3 can be used

- $\Pr[(\Delta U, \Delta V) = (0^n, 0^t)] \leq \frac{1}{2^{n+t}}$
- $\Pr[(\Delta U, \Delta V) = (0^n, 0^t)] \leq \frac{4}{2^{n+t}}$ for all cases
Instantiation Updates*

• In [IMPS17], we used Deoxys-BC and SKINNY to instantiate ZMAC
  – standard TPRP security assumption
• “XOR some extra tweak material to the key input of the TBC”
  – originally proposed by [LRW02] for BCs
• Given $\widetilde{E}^i : \{0, 1\}^k \times \{0, 1\}^t \times \{0, 1\}^n \to \{0, 1\}^n$, regard it as

$$E^i : \{0, 1\}^k \times \{0, 1\}^{t+k} \times \{0, 1\}^n \to \{0, 1\}^n$$

* Thanks to Christof Beierle for the suggestion.
Instantiation Updates

- **Input:** \( X = (X[1], \ldots, X[m]) \), 
  \(|X[i]| = n + (t + k), X[i] = (X_\ell[i], X_r[i]): X_r[i] \text{ is } t + k \text{ bits}

- **Output** \((U, V), |U| = n, |V| = t + k\)

- can process \((n + t + k)\) bits per 1 TBC call
Remarks

- related-key security of $\tilde{E}$ is needed (strong assumption)
- limited to the birthday security w.r.t. $k$
  - due to a generic birthday attack against $E_{K \oplus T}(\cdot)$ by [BK03]
  - $E_{K_i}(X)$ for $1 \leq i \leq 2^{k/2}$ and $E_{K \oplus T_j}(X)$ for $1 \leq j \leq 2^{k/2}$
- with Deoxys-BC-256, $k = 128$, $t = 124$, $n = 128$ (4 bits for domain separation)
  - 64-bit security, expected to be 50% faster
  - related-key security will not be an issue (also for SKINNY)
Instantiation with AES-128

- Can use ZMAC with AES-128
  - 64-bit security
  - estimated speed: 0.45 cpb (taking into account the 1.4 slowdown for recomputation of the key schedule at every block)
  - AES-256 is not suitable because of the related-key attack [BKN09] schedule)
Concluding remarks

• Reviewed ZMAC, a highly secure and fast MAC based on TBC
• Security Proof
• Instantiation updates

The power of XEX-like masking:

• We already see it in many blockcipher modes (e.g. PMAC, OCB)
• ZMAC shows it is also powerful for TBC modes
• As dedicated TBCs are becoming popular, this direction looks worth to be further explored
Concluding remarks

- Reviewed ZMAC, a highly secure and fast MAC based on TBC
- Security Proof
- Instantiation updates

The power of XEX-like masking:
- We already see it in many blockcipher modes (e.g. PMAC, OCB)
- ZMAC shows it is also powerful for TBC modes
- As dedicated TBCs are becoming popular, this direction looks worth to be further explored

Thank you!